

In the Claims:

Please amend Claims 1, 2, 5, 9-10, 12, 13, 16, 20-21, and 27-30, and cancel Claims 3-4, 11, 14-15, and 22, all as shown below.

1. (Currently Amended): A system for high availability clustering of a group of computer nodes, comprising:

A Java-based cluster server ~~computer~~ that allows an software application to access a set of resources of various resource types, including different software application servers, within a cluster, wherein said resources and software application servers are available at ~~said computer or at another~~ one or more computers in the cluster, and wherein the resources and software application servers can be grouped by resource type within a pool of resources;

~~a cluster server that operates at said computer and that allows access to said set of resources;~~

a resource interface provided by said Java-based cluster server that provides an abstraction layer and allows the Java-based cluster server to receive uniform requests from the application and communicate the requests to said set of resources;

a plurality of plugins that are plugged into the resource interface to provide a set of application-specific callbacks from the Java-based cluster server to the different resources, wherein the system includes a plugin for each resource type, and wherein each plugin implements a resource API to encapsulate its particular resource type-specific behavior and to isolate the Java-based cluster server from that behavior while providing access to its pool of resources;

wherein a JNDI interface provides an interface between the Java-based cluster server and a JNDI-compliant database;

wherein additional plugins may be plugged into the resource interface for other resource types; and

wherein the system can be extended by adding additional computers with Java-based cluster servers and resource interfaces operating thereon.

2. (Currently Amended): The system of claim 1 wherein each of said Java-based cluster servers includes a heartbeat interface that provides heartbeat information to other Java-based cluster servers at said other software application servers.

3 – 4. (Canceled)

5. (Currently Amended): The system of claim 1 wherein the system includes a cluster administration utility for accessing and administering the Java-based cluster server using remote method invocation calls.

6. (Original): The system of claim 1 wherein each resource has a resource type associated with it.

7. (Original): The system of claim 6 wherein resources are the object instances of their respective resource types.

8. (Original): The system of claim 1 wherein a resource is any of a computer, internet protocol address, disk, database, or file system or application.

9. (Currently Amended): The system of claim 1 wherein the Java-based cluster server defines resource groups that includes clusters of resources.

10. (Currently Amended): The system of claim 1 wherein the plugins include a WebLogic plugin and a Tuxedo plugin.

11. (Canceled)

12. (Currently Amended): A method for providing a high availability clustering framework system for a group of computer nodes, comprising the steps of:

allowing an software application to access, via a computer and a Java-based cluster server operating thereon, a set of resources of various resource types, including different software application servers, within a Java-based cluster wherein said resources are available at said computer or at another computer, and wherein the resources and software application servers can be grouped by resource type within a pool of resources;

providing a resource interface at said Java-based cluster server that provides an abstraction layer and allows the Java-based cluster server to receive uniform requests from the Java application and communicate the requests to said set of resources via a plurality of plugins that are plugged into the resource interface;

wherein the plurality of plugins are plugged into the resource interface to provide a set of application-specific callbacks from the Java-based cluster server to the different resources, wherein the system includes a plugin for each resource type, and wherein each plugin implements a resource API to encapsulate its particular resource type-specific behavior and to isolate the Java-based cluster server from that behavior while providing access to its pool of resources;

wherein a JNDI interface provides an interface between the Java-based cluster server and a JNDI-compliant database;

wherein additional plugins may be included in the resource interface for other resource types; and

wherein the system can be extended by adding additional computers with Java-based cluster servers and resource interfaces operating thereon.

13. (Currently Amended): The method of claim 12 wherein said Java-based cluster server includes a heartbeat interface provides heartbeat information to other Java-based cluster servers at said other software application servers.

14 - 15. (Canceled)

16. (Currently Amended): The method of claim 12 wherein the system includes a cluster administration utility for accessing and administering the Java-based cluster server using remote method invocation calls.

17. (Original): The method of claim 12 wherein each resources has a resource type associated with it.

18. (Original): The method of claim 17 wherein resources are the object instances of their respective resource types.

19. (Original): The method of claim 12 wherein a resource is any of a computer, ip address, disk, database, or file system or application.

20. (Currently Amended): The method of claim 12 wherein the Java-based cluster server allows for clustering resources within a resource group.

21. (Currently Amended): The method of claim 12 wherein the plugins include a WebLogic plugin and a Tuxedo plugin.

22-26. (Canceled)

27. (Currently Amended): A system for high-availability clustering in a network, comprising:

a Java-based cluster server computer that receives requests from a client software application to access one or a plurality of software application servers of different types within a cluster, wherein the software application servers are available ~~either at the computer or at another~~ one or more computers within the cluster;

~~a cluster server that operates at the computer and provides access to the plurality of application servers,~~ wherein the Java-based cluster server further comprises a resource interface that provides an abstraction layer and allows the Java-based cluster server to receive uniform requests from the client application and communicate the requests to the software application servers; and

a plurality of plugins that can be plugged into the resource interface to provide application-specific callbacks from the Java-based cluster server to the different software application servers, wherein the system includes a plugin for each software application server type, and wherein each plugin implements a resource interface that encapsulates the particular resource type-specific behavior for that software application server type, and isolates the Java-based cluster server from that behavior while still providing access to the software application server.

28. (Currently Amended): The system of claim 27 wherein, for each software application server type, an appropriate plug-in is loaded at the time the first software application server of a

defined type is created, and wherein a handle is created to the specific resource instance, which can then be used by the Java-based cluster server in subsequent method calls.

29. (Currently Amended): A method for high-availability clustering, comprising the steps of:

receiving requests at a Java-based cluster server ~~computer~~ from a client software application to access one or a plurality of software application servers of different types within a cluster, wherein the software application servers are available either ~~at the computer or~~ at another one or more computers within the cluster;

communicating the requests to a Java-based cluster server that operates at the computer and provides access to the plurality of software application servers, wherein the Java-based cluster server further comprises a resource interface that provides an abstraction layer and allows the Java-based cluster server to receive uniform requests from the client application and communicate the requests to the software application servers; and

using a plurality of plugins that can be plugged into the resource interface to provide application-specific callbacks from the Java-based cluster server to the different software application servers, wherein the system includes a plugin for each software application server type, and wherein each plugin implements a resource interface that encapsulates the particular resource type-specific behavior for that software application server type, and isolates the cluster server from that behavior while still providing access to the software application server.

30. (Currently Amended): The method of claim 29 wherein, for each software application server type, an appropriate plug-in is loaded at the time the first software application server of a defined type is created, and wherein a handle is created to the specific resource instance, which can then be used by the Java-based cluster server in subsequent method calls.